

jc605 U.S. PTO
06/26/98

PATENT APPLICATION TRANSMITTAL LETTER

Docket Number: SWA-04

jc51 U.S. PTO
06/26/98

TO THE COMMISSIONER OF PATENTS AND TRADEMARKS:

Transmitted herewith for filing is the patent application of Darin Ellis and Michael Doyle for A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS.

Enclosed are:

12 sheets of drawings. (formal)

an assignment of the invention to JUXTACOMM TECHNOLOGIES, INC.

a certified copy of a _____ application.

Combined Declaration and Power of Attorney

a Verified Statement to establish small entity status under 37 CFR 1.9 and 1.27.

CLAIMS AS FILED

	NUMBER FILED	NUMBER EXTRA	RATE	Fee
BASIC FEE		0	\$790	\$790
TOTAL CLAIMS	18 -20= *	0	x\$22	0
INDEPENDENT CLAIMS	4 -3= *	1	x\$82	82
MULTIPLE DEPENDENT CLAIM PRESENT			\$270	
*Number extra must be zero or larger			TOTAL	\$872
<input checked="" type="checkbox"/> If applicant is a small entity under 36 CFR 1.9 and 1.27, then divide total fee by 2, and enter amount here.			SMALL ENTITY TOTAL	\$436

A check in the amount of \$436.00 to cover the filing fee is enclosed.

The Commissioner is hereby authorized to charge and credit Deposit Account No. 08-0719 as described below. I have enclosed a duplicate copy of this sheet.

Charge the amount of \$ _____ as filing fee

Credit any overpayment.

Charge any additional filing fees required under 37 CFR 1.16.

for this filing, and/or

for later presentation of extra claims.

Charge any patent application processing fees under 37 CFR 1.17.

for this filing only, or

during pendency of this application.

Charge the issue fee set in 37 CFR 1.18 at or before mailing of the Notice of Allowance, pursuant to 37 CFR 1.311(b).

I hereby certify that this paper or fee is being deposited with the U.S. Postal Service "Express Mail" Post Office to Addressee" service under 37 CFR 1.10 on the date indicated below and addressed to the Commissioner of Patents and Trademarks, Washington, DC 20231.

Brian J. Hamilla
DATE: 06/26/98
EXPRESS MAIL NO. EE243336603US

June 26, 1998
Date


Signature
Brian J. Hamilla
Registration No. 38,482

**A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA BETWEEN
DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS**

FIELD OF THE INVENTION

5 This invention relates to a system and method for importing, transforming and exporting data between distributed heterogeneous computer systems and in particular to a system of script processing utilizing metadata to control data transformation within the system
10 and data movement into and out of the system.

BACKGROUND OF THE INVENTION

15 Data exchange between distributed heterogeneous computer systems has been problematic in the industry. Businesses frequently use disparate data formats and data storage types within a corporate structure. As well, business partners almost invariably use different data formats. To permit data exchange when different formats are used, a static inter-communication facility must be
20 maintained for each pair of disparate data formats and/or data storage types. Changes to data formats or data storage types force the re-engineering of the corresponding facility.

25 A data import/export system is taught in United States Patent No. 5,497,491 which issued on March 5, 1996 to Mitchell et al. That patent describes a system and method for importing and exporting data between an external object oriented computing environment. The system and method requires a datalist object for each
30 field to be moved from the external object oriented computing environment to the external computing environment. A metadata object is required for each datalist object. The system is therefore complex and resource-use intensive. Furthermore, it is only capable
35 of moving data from an object-oriented to some other

computer environment. The system is therefore inflexible and unsuitable for use in many applications where import/export must be performed between two computer systems that do not use object oriented data formats.

5 Therefore, what is needed is a distributed system and method that is capable of transforming data from a source computer system into data usable by a computer system which stores data in a different format. This system must provide a simple means for specifying the
10 transformation definitions and for controlling the flow of data from an input data source to an output data target. Configuration management of the system must be dynamic to respond to the changing business environment and non-intrusive to minimize the effects of changing data formats
15 or data storage types.

SUMMARY OF THE INVENTION

It is therefore an object of this invention to provide a system and method for data transformation and
20 data exchange between distributed heterogeneous computer systems.

It is another object of the present invention to provide a script processing language that defines operations to control data transformation within the
25 system and data movement into and out of the distributed system, utilizing metadata definitions.

It is another object of the present invention to provide a format control language that defines the transformation of an external data source into data bags
30 and of the internal data bags to an external data target.

It is another object of the present invention to provide a means of configuration management that allows a user of the system to define scripts, import data connections, export data connections, data bags, and rule
35 set definitions and to store them in a metadata database.

It is another object of the present invention to provide a means of executing scripts in order to control the distributed transformation system.

According to the invention, there is provided a system for transforming and exchanging datastore data between heterogeneous computer systems using different datastore formats for storing similar information, the system comprising: means for transforming and processing import datastore data into generic format data according to predetermined import transformation rules and functions; means for converting the generic format data into export datastore data according to predetermined export transformation rules and functions; and interface to communications means for receiving the import datastore data and for transmitting the export datastore data.

A datastore refers to the storing of any type of data in a persistent storage system, such as on magnetic media like a disk drive. The types of data stored could include text or binary.

As will be shown below, the present invention can be used to create import data definitions, data bag storage, data bag transformation definitions or rule sets, export data definitions and scripts to control the usage of all those definitions in the process of transforming and exchanging data between dissimilar computer systems.

A generic format data bag contains both the data to be manipulated and the data structure definitions, in a generic format. The present invention will use the title 'data bag' to indicate a generic format data bag.

30

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a block diagram of a data transformation and exchange environment, an external distributed computing environment and the associated hardware platforms.

FIG. 2 shows a block diagram of a system for transforming and exchanging data between heterogeneous distributed computing environments according to the present invention.

5 FIG. 3 shows the components of the present invention that are defined within the metadata database.

FIG. 4 shows the operations performed by the import data interface 32 and the export data interface 34 when the script processor 37 of FIG. 2 is invoked.

10 FIG. 5 is a flow diagram showing operations performed by the configuration management user interface 39 of FIG. 2 at program execution time.

FIG. 6 is a flow diagram showing operations performed by the script processor 37 of FIG. 2.

15 FIG. 7 shows an example of the operations to define the components for a data transformation.

FIG. 8 shows an example script to control the data transformation defined in FIG. 7.

20 FIG. 9 shows an example of part of a rule that could be used in the data transformation defined in FIG. 7.

FIG. 10 shows the internal storage of an example ODBC-enabled database table used in the data transformation defined in FIG. 7.

25 FIG. 11 shows the internal storage of the data bag used to store the imported data defined and used in the data transformation defined in FIG. 7.

FIG. 12 shows the internal storage of the data bag used to store the data for export that is defined and used in the data transformation defined in FIG. 7.

30 FIG. 13 shows the internal storage of the export data target used in the data transformation defined in FIG. 7.

FIG. 14 shows the data layout, such as might 35 appear in a computer program, of a text file containing

personal information records. There is a repeating group of information at the end of each record. This data layout example will be used to show how data bags can handle repeating groups of data.

5 FIG. 15 shows the text file, defined in FIG. 14, with some example data.

10 FIG. 16 shows a data bag containing the data from the text file defined in FIG. 15. The data group definition 162 shows how the 'CHILDREN' group is defined and the data group collection 163 shows how the 'CHILDREN' group is stored.

15 FIG. 17 shows an example rule that would act on the data bag defined in FIG. 16 and output only the personal records that contained children whose age is less than 20.

DETAILED DESCRIPTION OF THE INVENTION

20 Prior to describing a system and method for data transformation and data exchange between distributed heterogeneous computer systems according to the present invention, a general overview of the computing environment will be provided. A general description of the system and method of the present invention will then be provided, followed by a detailed design description for the system 25 and method for data transformation and data exchange according to the present invention.

30 Referring to FIG. 1 and FIG. 2, the hardware and software environment in which the present invention operates will now be described. The present invention is a method and system for data transformation and data exchange between an external distributed computing environment 12 operating on one or more computer platforms 11 and a transformation/exchange system 13 operating on one or more computer platforms 14. It will 35 be understood by those having skill in the art that each

of computer platforms 11 and 14 typically include computer hardware units such as main memory 17, a central processing unit (CPU) 18 and an input/output (I/O) interface 19, and may include peripheral components such as a display terminal 21, an input device such as a keyboard 22 or a mouse 23, nonvolatile data storage devices 24 such as magnetic or optical disks and other peripheral devices. Computer platform 11 or 14 also typically includes microinstruction code 16, and an operating system 15. As one example, each computer platform 11 and 14 may be a desktop computer having an IBM PC architecture. Operating system 15 may be a Microsoft Windows NT operating system. FIG. 2 is a functional block diagram of the current invention. It will be understood by those having skill in the art that this architecture might be implemented on multiple machines and will vary according to the application.

Referring to FIG. 1, a system 13 for transformation and exchange between distributed heterogeneous computer systems 12, according to the present invention, is shown. As shown in FIG. 2, the transformation and exchange system 13 includes an import data interface 32 to import data from an import data source 31 into the transformation and exchange system 13. As shown in FIG. 4, the import data interface 32 includes an import data connection 41, an import data view 42 of the import data source 31 and a generic format data bag 43 where the imported data is to be stored. Those skilled in the art will understand that a view is a logical subset of the content of an actual external data source. The import data view 42 is a logical subset of the content of the import data source 31. As will be shown below, the import data view 42 will be used during the execution of the script processor 37 (FIG. 2) to load data from the import data source 31 into the data bag 43.

Data bags **43** are used in the present invention for the storage and transformation of external data. A data bag contains both the definition of the data contained within the data bag and the actual generic format data. Generic format data refers to data that has been stored within the present invention and is now independent of the original data source. Data stored in this generic format can be transformed into any required format for exporting to an export data target **33** (FIG. 2). Data bags are stored in non-persistent storage, like main memory **17**, are created by the script processor and exist while the script is running. Data bags can contain fixed format data, data grouping and repeating data groups.

15 Referring again to FIG. 2, the system includes
an export data interface 34 to export a data bag 44 out to
an export data target 33. As shown in FIG. 4, the export
data interface 34 includes generic format data bag 44
where data for exporting is stored, the export data
view 45 of the data bag 44 and the export data
connection 46. As will be shown below, the export data
view 45 of the data bag 44 will be used during the
execution of the script processor 37 to save data from the
data bag 44 out to the export data target 33.

25 Also shown in FIG. 2 and FIG. 3, the system includes a configuration management user interface 39 to define the components of the present invention, which include external data connections 51, views 52, data bags 53, rule sets 54 and scripts 55. These component 30 definitions are stored in the metadata database 38. The data bags are stored in the internal datastore 35. The component definitions will be described in detail below.

The transformation and exchange system 13
includes a script processor 37, in order to run scripts 55
35 defined in the metadata database 38. The script

processor 37 identifies the script command and invokes the correct method for that script command. The transformation/exchange system 13 also contains a rule processor 36 that is invoked by the script processor 37 to 5 transform one data bag into another data bag based on a rule. Rules will be described below.

FIG. 5 is a flow diagram showing the components that can be defined using the configuration management user interface 39 and the actions taken when defining each 10 component.

Connections 51 must have their connection type and properties defined. The connection type will be any of the industry standard data storage types, such as ODBC-enabled databases, spreadsheets, message-oriented 15 middleware and text files. The properties will include the name and location of the external data storage.

Views 52 must be associated with either an import data connection 41 (FIG. 4) or an export data connection 46. Each data connection has one or more views 20 of the external data. These views are used to import different collections of data from an import data source 31 (FIG. 2), or to export different collections of data from an export data bag 44 out to an export data target 33.

25 Data bag definition 53 contain two types of data collection: a data definition collection and a data group collection. A collection is a logical grouping of records that use the same format method. All the element 30 definitions for a data bag are stored within the data definition collection. Each row of data in a data bag is stored as one data group in the data group collection. The data group collection contains all the data groups in 35 the data bag. An import data connection 41 must have one or more import data views 42 and each data view must have an associated import data bag 43. Using the data

definitions in the data definition collection 112 (FIG. 11) of a data bag, the import data view 42 of the import data connection 41 is loaded in the import data bag 43. An export data connection 46 must have one or 5 more export data views 45 and each data view must have an associated export data bag 44. Using the data definitions in the data definition 112 of a data bag, the export data view 45 of the export data connection 46 is written using the data contained in the data bag. Data bags are also 10 defined for use by script commands that require import and export data bag(s), where these commands transform the data from the import data bag 43 and place the results in the export data bag 44.

Rule sets 54 (FIG. 5) are collections of rules 15 within the present invention. Rule sets are used to transform a data bag in one format into another data bag of a different format. The purpose of a rule is to perform a specific operation to achieve a desired result. A rule is one or more statements. These statements are 20 executed from top to bottom and when the last statement within the rule has been executed, or an Exit statement is encountered, the rule ends.

A statement is a single line in a rule. The types of statements implemented by the present invention, 25 within the rule set processor, includes comments, conditional processing, exiting a rule, looping, variable declaration and variable assignment.

Conditional processing, looping and assignment statements contain expressions. Elementary expressions 30 include strings, numbers, content of a variable and return value of a function. Functions are categorized into character manipulation, string manipulation, including other rules, initialization information, external file manipulation, variable content reporting and user 35 interface.

Complex expressions combine many elementary expressions in some manner, for the purpose of producing a single result. Complex expressions can be either arithmetic or conditional.

5 Complex arithmetic expressions are numeric elementary expressions that are combined to produce a single arithmetic result. Such expressions follow the standard format of all numeric expressions. Numbers are acted upon by numeric operators such as addition,
10 subtraction, multiplication, division, modulo and exponential. Brackets are used to group numbers and operators which need to be evaluated together.

15 Conditional expressions return the value True or False. These types of expressions are used to control conditional processing within the rules. Brackets are used to group conditions which need to be evaluated together. Complex conditional expressions are formed by combining simple conditions with 'And' or 'Or' operators.

20 Simple conditions have a 'left side' 'operator' 'right side' format. The left and right sides are elementary expressions. The logical operators that can be used for these conditions are equals, greater, less, not equal, greater or equal, less or equal, 'like' and 'in'. A simple condition can be negated by using the word 'not' 25 in front of the condition.

Scripts 55 must be defined to control data movement into and out of the system, and to control data transformation within the system.

30 FIG. 6 is a flow diagram showing the actions taken by the script processor 37 of the present invention.

35 The LOAD command permits an import data view 42 to be used to load data from an import data source 31 into an import data bag 43. The import data view 42 is associated with an import data connection 41, which specifies the import data source 31.

The SAVE command permits an export data view 45 to be used to save data from an export data bag 44 out to an external data target 33. The export data view 45 is associated with an export data connection 46, which 5 specifies the external data target 33.

The MERGE command permits two or more specified data bags, of the same data bag type, to be merged into another data bag. Only non-duplicate data groups from the input data bags are merged into the output data bag.

10 The JOIN command permits two or more specified data bags to be joined together into another data bag, dependent on the matching of a specific key value.

15 The APPEND command permits one data bag to be appended to the end of another data bag of the same data bag type.

The COPY command permits one data bag to be copied to another specified data bag of the same data bag type. If the target data bag exists, it will be overwritten.

20 The FORMAT command permits the transformation of a specified data bag into another data bag, possibly of a different data bag type, using a specified rule. This command will invoke the rule processor 36 to take the input data bag, transform the data according to the rule 25 statements and populate the output data bag with the transformed data.

The SORT command permits a data bag to be sorted by one or more data elements within the data bag. Each element can have an ascending or descending sort applied 30 to it. The result can be placed back into the original data bag or the result can be written to another data bag of the same type.

35 An example of a transformation according to a preferred embodiment of the invention will be shown in the following description. The embodiment will describe the

definition and usage of an import data connection, export data connection and transformation requirements to convert an Open Database Connectivity (ODBC) enabled database table into a delimited flat file. This example was chosen 5 because of the widespread usage of both ODBC and delimited files in the business community and of the direct application of the invention to the problem of transforming data between these two standards used by heterogeneous computer systems.

10 FIG. 7 shows the steps taken by the user to set up the definitions required for the preferred embodiment and to initiate the script to carry out the transformation from ODBC database data to delimited data. The user will invoke the configuration management user interface 39 15 (FIG. 2) to create the required definitions.

As shown in step 72, the user defines the import data connection 41 with a connection type of ODBC-enabled database, defines the location of the import data source 31 and defines the table within the database to be 20 used as the data source. In step 73 the user defines the import data bag to hold the data imported from the import data connection 41. In step 74 the import data view 42 is created to define the fields within the import database table that are to be used when processing the data source. 25 The import data view 42 is associated with the import data bag 43 that will receive the incoming data.

In step 75, the export data connection 46 is defined with a connection type as file and defines the location of the target data file is defined. In step 76, 30 the user defines the export data bag that will hold the data to be used by the export data connection 46. In step 77 the export data view 45 is created to define the layout of the target data file. The export data view 45 is associated with the export data bag 44 that will be 35 used to send data to the external data target 33.

In step 78, the rule definition allows the user to specify a complex set of statements to control the transformation of one data bag to another data bag. The statements in the rule come from the format control language which includes conditional logic flow control, looping and the ability to define and call functions not defined within the language.

In step 79, the user then defines the script that will load the import data source 31 into an import data bag 43, transform the loaded import data bag 43 into an export data bag 44 by executing the rule processor 36 using the specified rule (Rule1), and then exports the export data bag 44 to the external data target 33.

Finally, in step 80, the user initiates the script processor 37 to execute the script. The script processor 37 can be initiated from the graphical interface or from an interface external to the system.

FIG. 8 shows the script defined for this example. The first script command 81 uses the import data connection 41 and import data view 42 to load the data from the import data source 31 into the import data bag 43. The second command 82 transforms the data bag 43 into an export data bag 44 using the specified rule set (RuleSet1). Once the export data bag 44 has been populated with the transformed data it can be saved 83 directly out to the export data target 33, using the export data view 45 and the export data connection 46.

FIG. 9 shows an example rule for this example. The example rule demonstrates the use of conditional flow control (IF statement), record selection based on incoming data content (IN.CITY = "OTTAWA") and data transformation using assignment statements (for example, OUT.NAME = APPEND(IN.FIRST_NAME, " ", IN.LAST_NAME)). In step 78 of FIG. 7, RuleSet1 is defined to contain one rule (Rule1) which transforms data bag MAILING_DBAG into data bag

CITY_DBAG. When Rule1 is executed in the example shown in FIG. 9, the import data bag refers to MAILING_DBAG and the export data bag refers to CITY_DBAG.

FIG. 10 shows an example import data source 31 for this example. The internal storage of an ODBC-enabled database table is shown. The data in this table will be used to illustrate the data transformation defined in FIG. 7. The import data connection 41, defined in step 72, refers to the exact location of the database file ADDRESS.MDB 101 and indicates that the database is ODBC-enabled. The import data view 42, defined in step 74, specifies that all the fields in the data source table will be imported into the import data bag 43, defined in step 73.

FIG. 11 shows the internal storage of the import data bag 43, defined in step 73, which is used in the data transformation in FIG. 7. The data definition collection 112 specifies the key name used for locating fields in the data group collection 113 and specifies the data type for a field value associated with each key. All the fields in the data source table have been imported into the MAILING_DBAG data bag 111. This import data bag is created by the LOAD script command in step 81, (FIG. 8) using metadata definitions from the metadata database 38.

FIG. 12 shows the internal storage of the export data bag 44, defined in step 76, which is used in the data transformation described with reference to FIG. 7. The data group definition 122 is different than the data group definition 112 shown in FIG. 11. The CITY_DBAG data bag 121 contains three of the original six fields from the MAILING_DBAG, the import 111 data bag, as well as a computed field that is a concatenation of the first and last names from the import data bag. The CITY_DBAG 12 export data bag is created by FORMAT script command in step 82, using metadata definitions from the metadata

database **38** (FIG. 2). FIG. 9 shows part of Rule1, which is contained in RuleSet1 and defined in step **78**. The rule set in this example filters out all data group collection records in the MAILING_DBAG import data bag that have a **5** city name of 'OTTAWA' and then writes those records into the CITY_DBAG export data bag.

FIG. 13 shows the internal storage of the export data target **33** for this example. The internal storage of a delimited flat file is shown. The export data **10** connection **46**, defined in block **75**, refers to the exact location of the flat file CITY.CSV and indicates that the file is delimited. The export data view **45**, defined in step **77**, specifies that all the fields in the data bag will be exported to the delimited flat file **131**, defined **15** in step **73**. The CITY.CSV flat file is created by the SAVE script command in step **83**, using metadata definitions from the metadata database **38**.

FIG. 14 shows a second import data example. The storage format of a personal information text file is **20** shown. Each record contains a group at the end of the record, with repeating information about children of the specified person. This file definition will be used to illustrate the data storage of repeating group information in a data bag and the rule processing of the repeating **25** group information during a data bag transformation. This file definition will be used to create the import data interface **32** used in this example.

FIG. 15 shows the internal storage of the text file defined in FIG. 14. Each record contains a common **30** set of fields before the 'CHILDREN' group. At the end of each record the 'CHILDREN' group may contain from zero to ten sets of 'child' information, consisting of the child's name and age. Each record is terminated by an end-of-record indicator appropriate to the computer system on **35** which the file resides.

FIG. 16 shows the internal storage of the import data bag 43, that contains the imported data of the text file shown in FIG. 15. The data definition collection 162 now shows an example of a 'group' item type. The 5 'CHILDREN' group is defined as containing two fields, as specified by the two entries following the 'CHILDREN' group entry. The data group collection 163 shows how each record from the import text file, shown in FIG. 15, is stored. The number of occurrences of the data group, 10 defined by 'NBR_CHILDREN', must be stored so that the correct number of sets of the 'CHILDREN' group can be processed when manipulating the import data bag.

FIG. 17 shows an example rule created to transform the REPEATING_DBAG import data bag defined in 15 FIG. 16. This rule is one rule of a rule set. The rule will output the parent name, child name and child age for each input child whose age is less than 20. This example shows how a repeating information group can be manipulated within a data bag.

20 In the drawings and specification, there have been disclosed typical examples of the use of a preferred embodiment of the invention. Although specific terms have been employed to describe the preferred embodiment, they are used in a generic and descriptive manner only and not 25 for purposes of limitation. The scope of the invention is set forth in the following claims.

WE CLAIM:

1. A system for transforming and exchanging datastore data between heterogeneous computer systems using different datastore formats for storing similar information, said system comprising:

means for transforming and processing import datastore data into generic format data according to predetermined import transformation rules and functions;

means for converting said generic format data into export datastore data according to predetermined export transformation rules and functions; and

an interface to communications means for receiving said import datastore data and for transmitting said export database data.

2. A distribution system for transforming and exchanging data between heterogeneous computer systems, comprising:

a) a systems interface for defining logical import and export data interfaces, data transformation rule sets and scripts;

b) a metadata database for storing said logical import and export data interfaces, data transformation rule sets and scripts;

c) a script processor for controlling data transformation within said systems interface and movement of said data into and out of said distribution system; and

d) a rule set processor responsive to said script processor for manipulating a data bag for storing imported data and a data bag for storing export data.

3. A distribution system as claimed in Claim 2, wherein said systems interface comprises a configuration management user interface used by a user to define said

logical import and export data interfaces, and create data transformation rule sets and scripts.

4. A distribution system as claimed in Claim 3, wherein said logical import and export data interfaces comprise import and export data connections, import and export data views and said import and export data bags.

5. A distribution system as claimed in Claim 4, wherein said logical import data interface is used to import data from an import data source into said distribution system.

6. A distribution system as claimed in Claim 5, wherein said import data view is used during execution of said script processor to load data from said import data source into said import data bag.

7. A distribution system as claimed in Claim 4, wherein said logical export data interface is used to export data in said data bag out to an export data target.

8. A distribution system as claimed in Claim 7, wherein export data view of said export data bag is used during execution of said script processor to save data from said export data bag out to said export data target.

9. A distribution system as claimed in Claim 2, wherein defined scripts stored in said metadata database are executed by said script processor.

10. A distribution system as claimed in Claim 2, wherein said rule processor is invoked by said script processor to transform the import data bag into the export data bag based on predefined data transformation rules.

00000000 00000000
11. A distribution system as claimed in Claim 2 wherein said script processor makes use of a script control language for controlling data transformation within said system interface and movement of said data into and out of said distribution system.

12. A distribution system as claimed in Claim 11, wherein said script control language comprises a set of script commands and a script command processor to process and execute each of a number of script command lines.

13. A distribution system as claimed in Claim 12, wherein said set of script commands comprises a load command to load data into an import data bag from an import data connection; a sort command for sorting data in a data bag into a different order; a merge command for merging together data in a number of data bags; an append command for appending data from one data bag into another data bag; a copy command for copying one data bag into another data bag; a join command for joining two or more data bags into another data bag; a format command for formatting a data bag into another data bag using a defined rule set; and a save command for saving data from an export data bag out to an export data connection.

14. In a distribution system for transforming and exchanging data between heterogeneous computer systems, a method of controlling data transformation within said distribution system, comprising the steps of:

a) loading data into an import data bag from a logical import data interface and performing any one or more of the following steps to convert the data to a desired format in an export data bag;

1) sorting said data according to a predetermined order;

2) merging data from a number of data bags into one data bag;

3) appending data from a first data bag into another data bag of the same type;

4) copying data from a first data bag into another data bag of the same type;

5) joining data from two or more data bags into another data bag using a specified key;

6) formatting data from a data bag into another data bag of a different type, using a defined rule set; and

b) saving the data in the export data bag out to an export data connection.

15. A method as claimed in Claim 14 wherein said logical import data interface comprises import data connections, import data views and said import data bag.

16. A method as claimed in Claim 15 wherein said logical import data interface is used to import data from an import data source into said distribution system.

17. A method as claimed in Claim 16 wherein said import data view is used during execution of a script processor to load data from said import data source into said import data bag.

18. A computer readable memory for transforming and exchanging datastore data between heterogeneous computer systems using different datastore formats for storing similar information, comprising:

means for transforming and processing import datastore data into generic format data according to

predetermined import transformation rules and functions; and

means for converting said generic format data into export datastore data according to predetermined export transformation rules and functions.

ABSTRACT

A system and method is described for importing data from a source computer system, manipulating and transforming of that data, and exporting the data to a target computer system under control of a script processor using stored metadata definitions. Metadata is used to describe the properties of the data being manipulated. The system includes a means for manipulating the metadata definitions. The metadata definitions are created to import data into the system, export data from the system, create views of the external data, store generic format data within the system, manipulate generic format data within the system and to control data flow through the system. Data is imported into the system using an import data definition to map the external data into an import data bag. Data imported into an import data bag becomes independent of the original data source. Data is manipulated within the system using script control commands and data is transformed within the system using rule sets that act upon data bags. Data is exported from the system using an export data definition to map the import data bag into the required export data bag format and then to write data in the export data bag to the external data target.

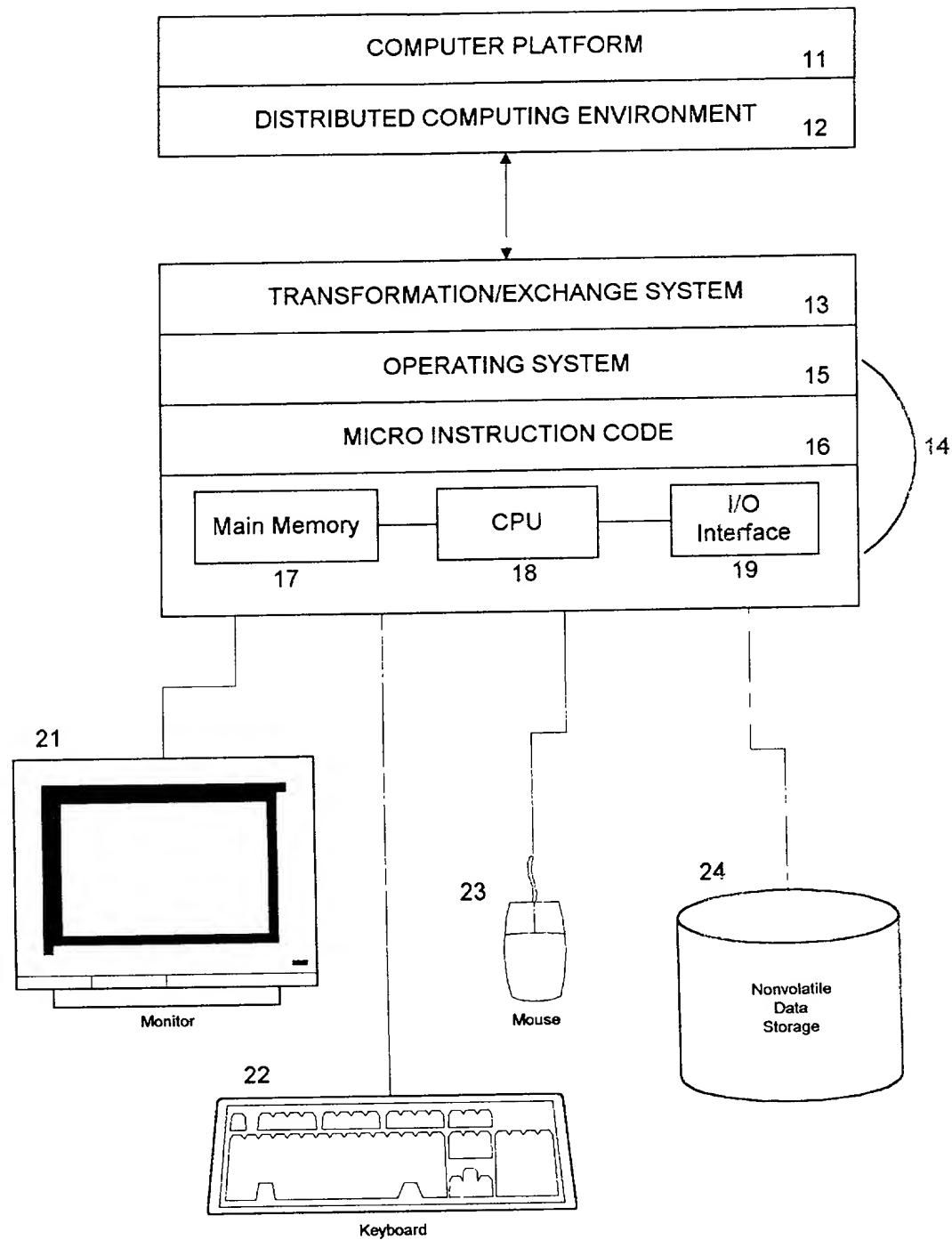


FIGURE 1

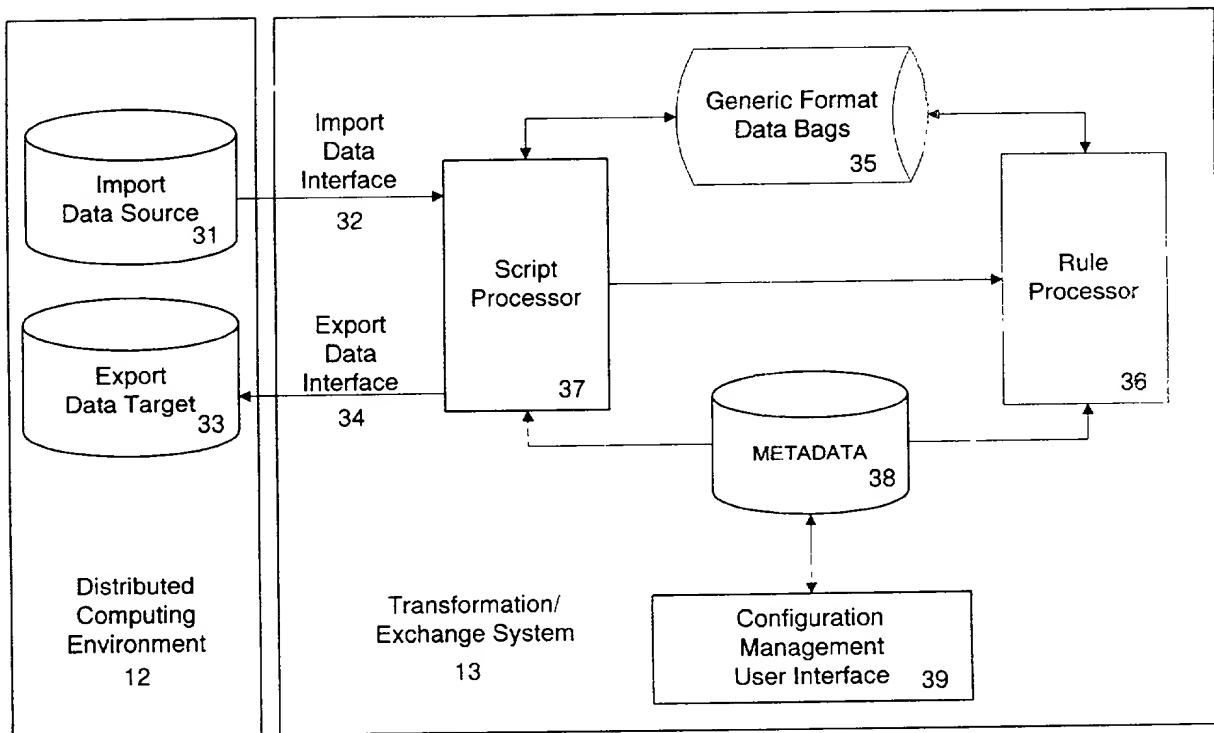


FIGURE 2

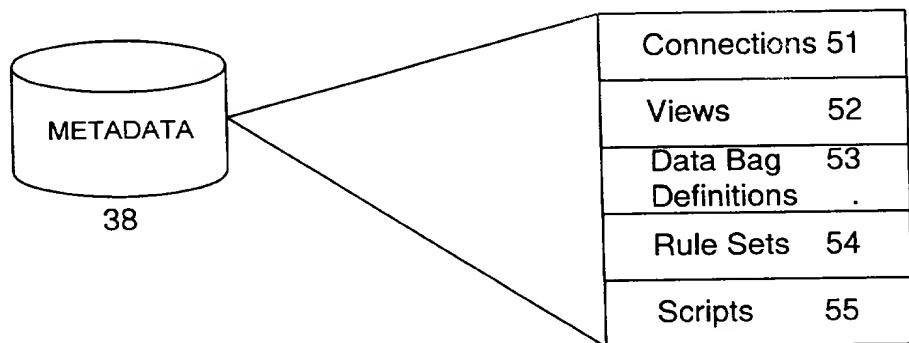


FIGURE 3

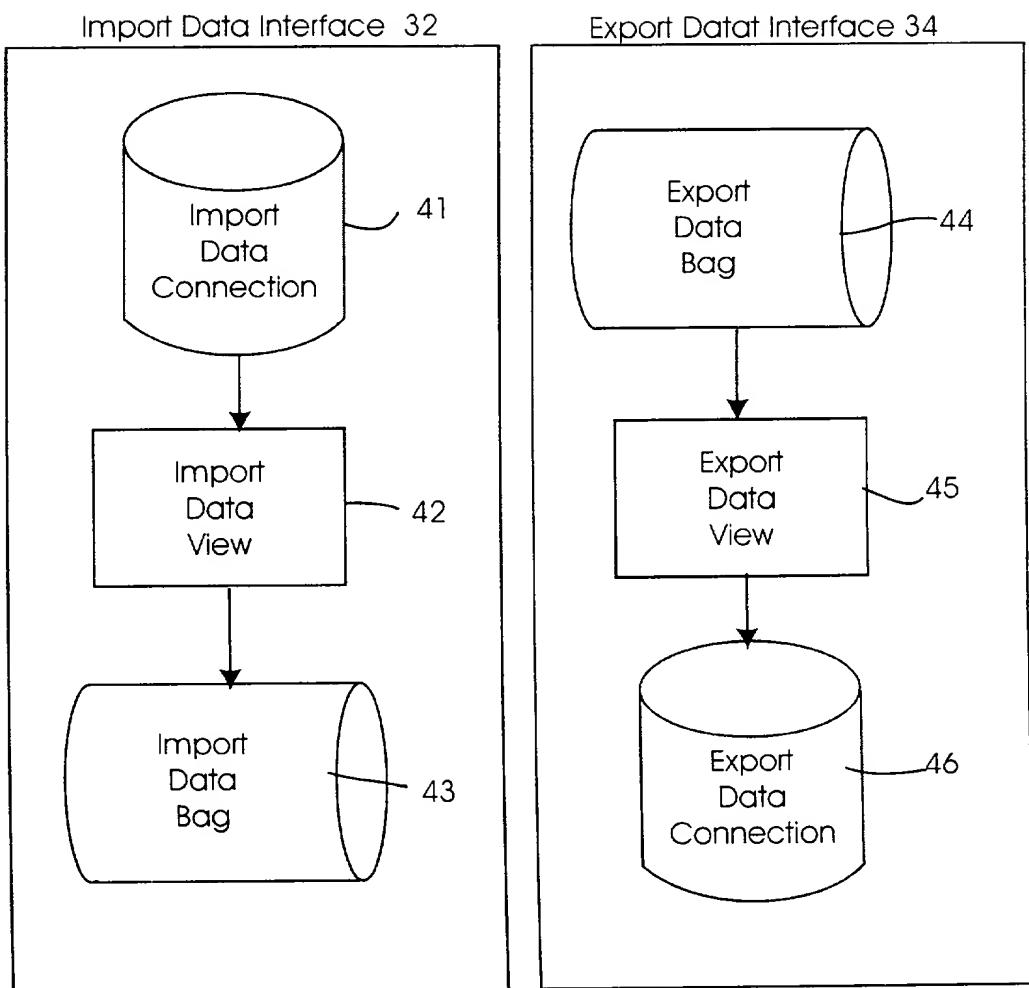


FIGURE 4

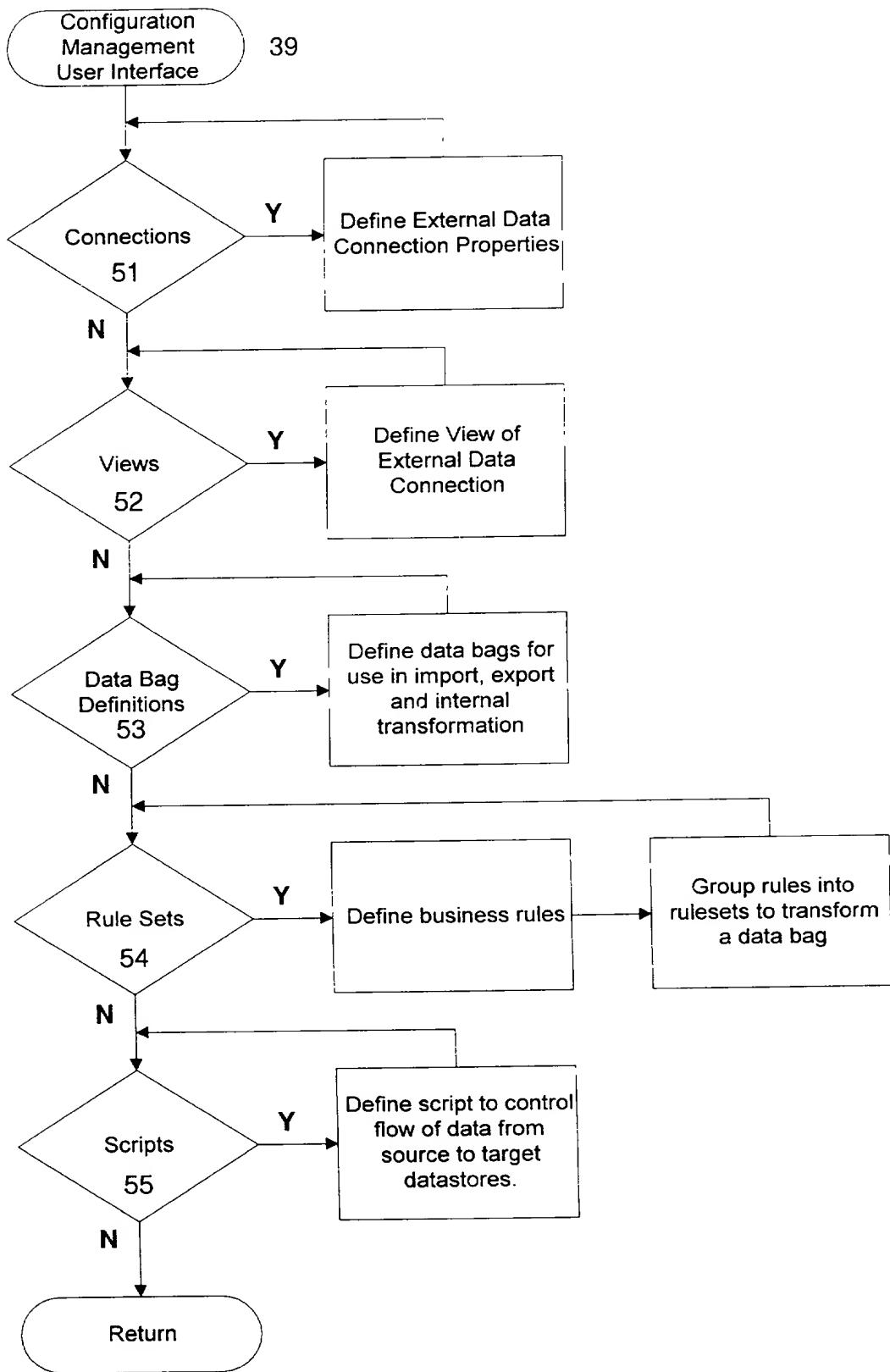


FIGURE 5

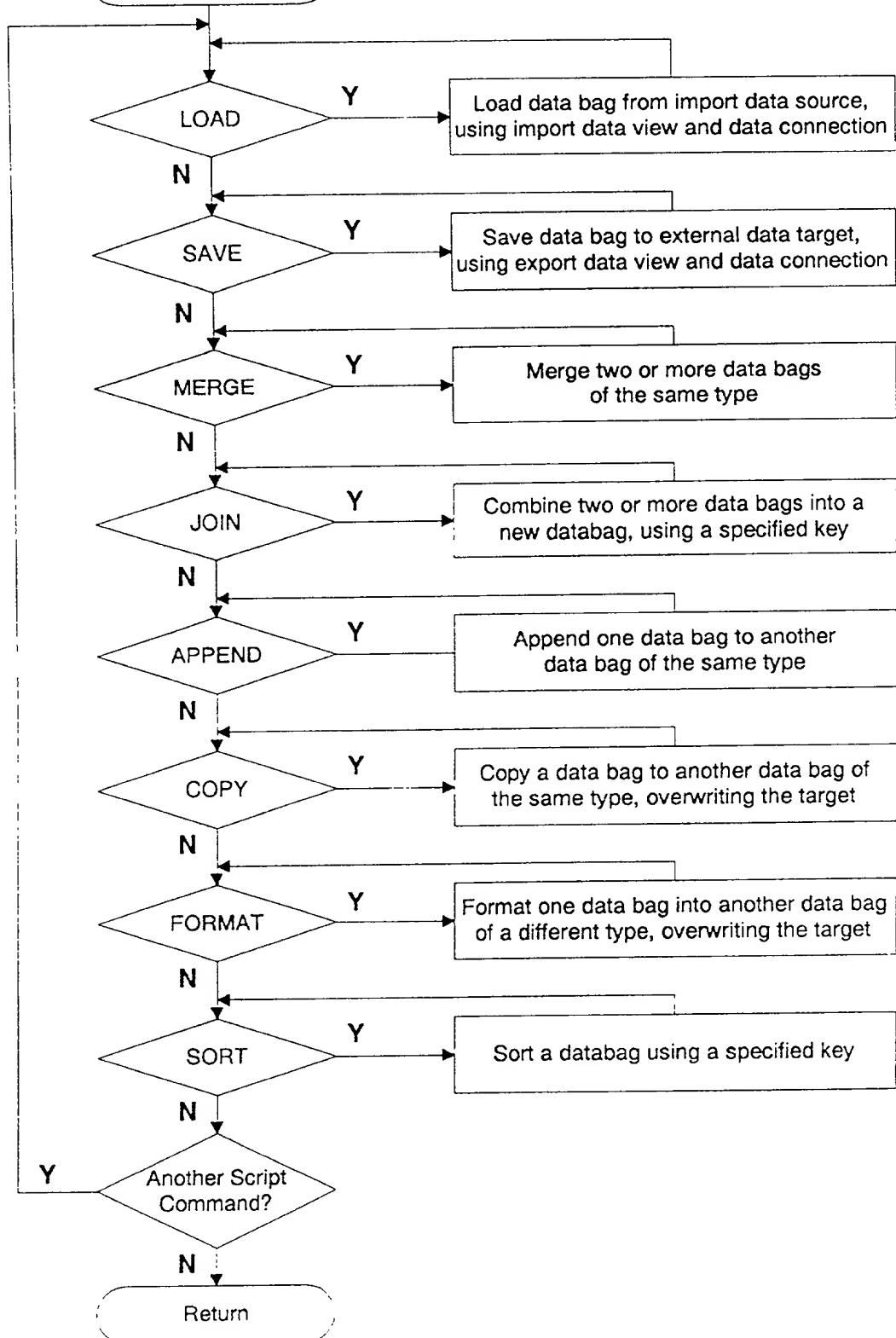


FIGURE 6

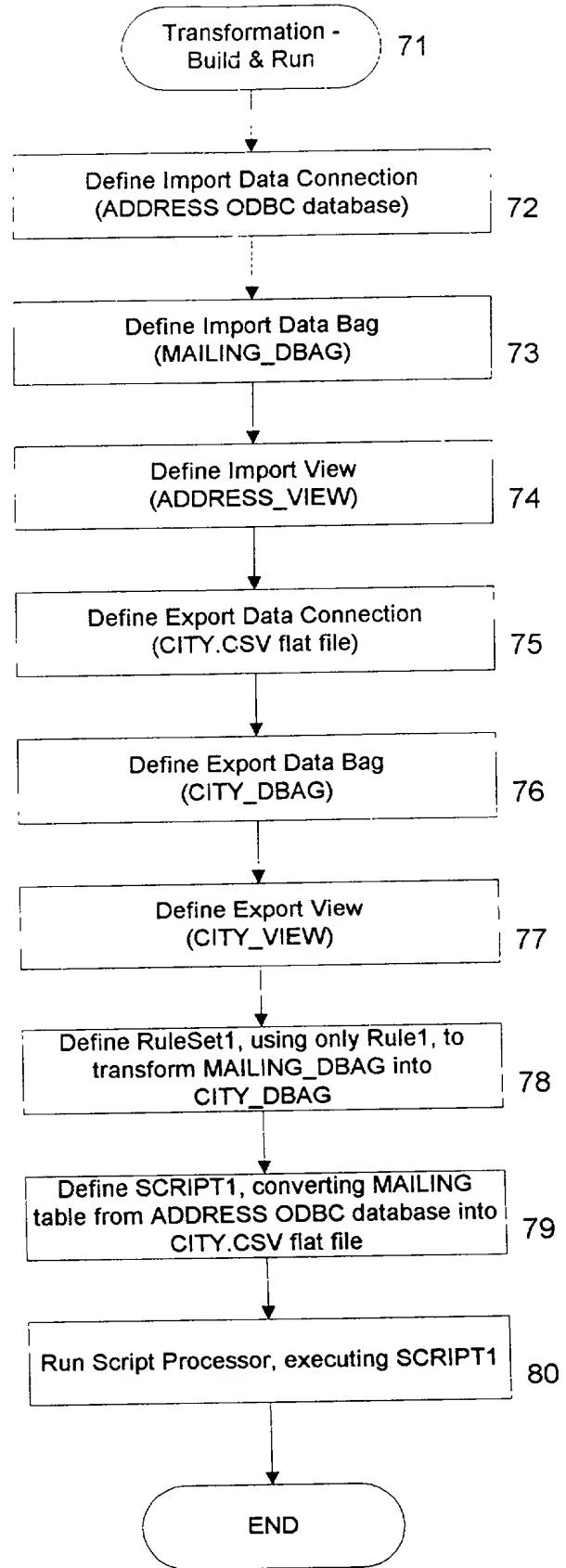


FIGURE 7

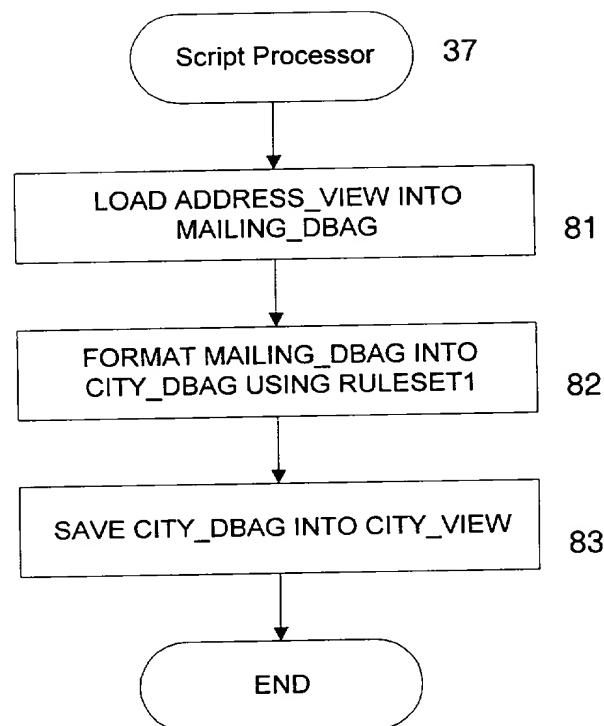


FIGURE 8

```
IF IN.CITY = "OTTAWA"  
    OUT.NAME = APPEND( IN.FIRST_NAME, " ", IN.LAST_NAME )  
    OUT.ADDR = IN.ADDR  
    OUT.CITY = IN.CITY  
    OUT.AGE = IN.AGE  
END IF
```

FIGURE 9

ODBC database : ADDRESS.MDB 101

FIRST_NAME	LAST_NAME	ADDRESS	CITY	COUNTRY	AGE
Bob	Drake	10 Penny St.	Toronto	CA	29
Susan	Hellman	55 Birch Rd.	Ottawa	CA	35
John	Bull	101 Pullman Ave.	New York	US	23
Josef	Karsch	31 Rideau St.	Ottawa	CA	85

FIGURE 10

Data Bag : MAILING_DBAG 111

DATA DEFINITION COLLECTION 112							
KEY	ITEM NAME	ITEM TYPE	PARENT	MAX OCCURS	ITEM COUNTER	LEVEL	
FIRST_NAME	FIRST_NAME			1		1	
LAST_NAME	LAST_NAME			1		1	
ADDRESS	ADDRESS			1		1	
CITY	CITY			1		1	
COUNTRY	PROV			1		1	
AGE	AGE			1		1	

DATA GROUP COLLECTION 113							
KEY	VALUE	KEY	VALUE	KEY	VALUE	KEY	VALUE
FIRST_NAME	Bob	FIRST_NAME	Susan	FIRST_NAME	John	FIRST_NAME	Josef
LAST_NAME	Drake	LAST_NAME	Hellman	LAST_NAME	Bull	LAST_NAME	Karsch
ADDRESS	10 Penny St	ADDRESS	55 Birch Rd.	ADDRESS	101 Pullman Ave.	ADDRESS	31 Rideau St.
CITY	Toronto	CITY	Ottawa	CITY	New York	CITY	Ottawa
COUNTRY	CA	COUNTRY	CA	COUNTRY	US	COUNTRY	CA
AGE	29	AGE	35	AGE	23	AGE	85

FIGURE 11

Data Bag : CITY_DBAG

121

DATA DEFINITION COLLECTION 122						
KEY	ITEM NAME	ITEM TYPE	PARENT	MAX OCCURS	ITEM COUNTER	LEVEL
NAME	NAME	TEXT		1		1
ADDRESS	ADDRESS	TEXT		1		1
CITY	CITY	TEXT		1		1
AGE	AGE	INTEGER		1		1

DATA GROUP COLLECTION 113			
KEY	VALUE	KEY	VALUE
FIRST_NAME	Susan	FIRST_NAME	Josef
LAST_NAME	Hellman	LAST_NAME	Karsch
ADDRESS	55 Birch Rd	ADDRESS	31 Rideau St
CITY	Ottawa	CITY	Ottawa
COUNTRY	CA	COUNTRY	CA
AGE	35	AGE	85

FIGURE 12

Delimited Flat File : CITY.CSV

131

Susan Hellman,55 Birch Rd,Ottawa,35
Josef Karsch,31 Rideau St,Ottawa,85

FIGURE 13

```

FIRST_NAME      CHAR(30)
LAST_NAME       CHAR(30)
ADDR_LINE1      CHAR(30)
ADDR_LINE2      CHAR(30)
CITY            CHAR(30)
PROV            CHAR(3)
NBR_CHILDREN    INTEGER(3)
CHILDREN        OCCURS 1 TO 10 TIMES
                  CHILD_NAME    CHAR(30)
                  CHILD_AGE     INTEGER(3)

```

FIGURE 14

Repeating Data File : REPEAT.TXT 151

Henry	Smith	61 Falldown Rd.	RR1	Greely	Ont	3	Amy	16	John	13	Bob	23
John	Kierstead	1034 Ayr Place	Apt 132	Ottawa	Ont	1	Susan	10				

FIGURE 15

Data Bag : REPEATING_DBAG 161

DATA GROUP DEFINITION 162

KEY	ITEM NAME	ITEM TYPE	PARENT	MAX. OCCURS	ITEM COUNTER	LEVEL
FIRST_NAME	FIRST_NAME	TEXT		1		1
LAST_NAME	LAST_NAME	TEXT		1		1
ADDR_LINE1	ADDR_LINE1	TEXT		1		1
ADDR_LINE2	ADDR_LINE2	TEXT		1		1
CITY	CITY	TEXT		1		1
PROV	PROV	TEXT		1		1
NBR_CHILDREN	NBR_CHILDREN	INTEGER		1		1
CHILDREN	CHILDREN	GROUP		1		1
CHILDREN[1].CHILD_NAME	CHILD_NAME	TEXT	CHILDREN	10	NBR_CHILDREN	2
CHILDREN[1].CHILD_AGE	CHILD_AGE	INTEGER	CHILDREN	10	NBR_CHILDREN	2

DATA GROUP COLLECTION 163

FIRST_NAME	Henry	FIRST_NAME	John
LAST_NAME	Smith	LAST_NAME	Keirstead
ADDR_LINE1	61 Falldown Rd.	ADDR_LINE1	1034 Ayr Place
ADDR_LINE2	RR1	ADDR_LINE2	Apt. 132
CITY	Greely	CITY	Ottawa
PROV	Ont	PROV	Ont
NBR_CHILDREN	3	NBR_CHILDREN	1
CHILDREN[1].CHILD_NAME	Amy	CHILDREN[1].CHILD_NAME	Susan
CHILDREN[1].CHILD_AGE	16	CHILDREN[1].CHILD_AGE	10
CHILDREN[2].CHILD_NAME	John		
CHILDREN[2].CHILD_AGE	13		
CHILDREN[3].CHILD_NAME	Bob		
CHILDREN[3].CHILD_AGE	23		

FIGURE 16

```
FOR CHILD_NDX = 1 TO NBR_CHILDREN
    IF CHILDREN[ CHILD_NDX ].CHILD_AGE < 20
        OUT.PARENT = APPEND( IN.FIRST_NAME, " ", IN.FIRST_NAME )
        OUT.CHILD_NAME = IN.CHILDREN[ CHILD_NDX ].CHILD_NAME
        OUT.CHILD_AGE = IN.CHILDREN[ CHILD_NDX ].CHILD_AGE
    END IF
END FOR
```

FIGURE 17

COMBINED DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

DECLARATION: As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled **A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS**, the specification of which

is attached hereto.
 was filed on _____ as U.S. Application Serial No. _____ or
on _____ as PCT Application No. _____ and amended on _____ (if
applicable.)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56

I hereby claim priority benefits under Title 35, United States Code §119(e) of provisional application for patent Ser. No. 60/051,052, filed 27 June 1997.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patents or inventor's certificate having filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Claimed

(Number)	(Country)	(Date)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Yes	No
(Number)	(Country)	(Date)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Yes	No
(Number)	(Country)	(Date)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Yes	No

I hereby claim the benefit under Title 35, United States Code, §120 of any United States Application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Appln. Ser. No.) (Filing Date) (Status)

(Appln. Ser. No.) (Filing Date) (Status)

POWER OF ATTORNEY: I hereby appoint the following attorneys, with full power of substitution, association, and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office concerned therewith.

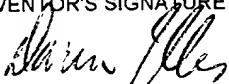
John B. Hardaway, III, Reg. No. 26,554; J. Bennett Mullinax, Reg. No. 36,221; Michael A. Cicero, Reg. No. 34,562; Brian J. Hamilla, Registration No. 38,482, and Katja Dunbar Peralta, Reg. No. P-40,768.

Address all correspondence to:

HARDAWAY LAW FIRM, P.A.
P.O. Box 10107, Fed. Sta.,
Greenville, South Carolina, 29603-0107,
United States of America

Telephone No.: (864) 233-6700

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

FULL NAME OF SOLE OR FIRST INVENTOR Darin ELLIS	INVENTOR'S SIGNATURE 	DATE 25 - JUNE - 98
RESIDENCE 812 Hauteview, Orleans, Ontario, Canada, K4A 2M1	CITIZENSHIP <input type="checkbox"/> Canadian	
POST OFFICE ADDRESS as above		
FULL NAME OF SECOND JOINT INVENTOR, IF ANY Michael DOYLE	INVENTOR'S SIGNATURE 	DATE 25 - JUNE - 98
RESIDENCE 75 Southpark Drive, Gloucester, Ontario, Canada, K1B 3B5	CITIZENSHIP <input type="checkbox"/> Canadian	
POST OFFICE ADDRESS as above.		

Applicant or Patentee: Darin Ellis et al

Serial or Patent No.: _____

Atty. Dkt. No. _____

Filed or Issued: _____

**FOR A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA
BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS**

86/92/90
C
S
P
O
L
D
09405501-002500

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) and 1.27(b)) - INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees under section 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office with regard to the invention entitled **A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA
BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS** described in

the specification filed herewith

application serial no. , filed

patent no. , issued .

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below.

no such person, concern, or organization persons, concerns or organizations listed below*

*NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

FULL NAME JUXTACOMM TECHNOLOGIES INC.

ADDRESS 75 Southpark Drive, Gloucester, Ontario, Canada, K1B 3B5

INDIVIDUAL SMALL BUSINESS CONCERN NONPROFIT ORGANIZATION

FULL NAME _____

ADDRESS _____

INDIVIDUAL SMALL BUSINESS CONCERN NONPROFIT ORGANIZATION

FULL NAME _____

ADDRESS _____

INDIVIDUAL SMALL BUSINESS CONCERN NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Darin ELLIS
NAME OF INVENTOR

Darin Ellis
Signature of Inventor

Michael DOYLE
NAME OF INVENTOR

Michael Doyle
Signature of Inventor

NAME OF INVENTOR

Signature of Inventor

Date 25-JUNE-98

Date 25-JUNE-98

Date

Applicant or Patentee: Darin Ellis et al

Serial or Patent No.: _____

Filed or Issued: _____

Title: A SYSTEM FOR TRANSFORMING AND EXCHANGING

DATA BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS

Atty's. Dkt. No.: _____

**VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) & 1.27(c)) - SMALL BUSINESS CONCERN**

I hereby declare that I am

the owner of the small business concern identified below:
 an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN: JUXTAComm TECHNOLOGIES INC.
ADDRESS OF SMALL BUSINESS CONCERN: 75 Southpark Drive, Gloucester, Ontario, Canada, K1B 3B5

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention, entitled A SYSTEM FOR TRANSFORMING AND EXCHANGING DATA BETWEEN DISTRIBUTED HETEROGENEOUS COMPUTER SYSTEMS by inventor(s) Darin Ellis et al described in

the specification filed herewith
 application serial no. _____ filed _____
 patent no. _____ issued _____

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e). *NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

NAME _____

ADDRESS _____

INDIVIDUAL SMALL BUSINESS CONCERN NONPROFIT ORGANIZATION

NAME _____

ADDRESS _____

INDIVIDUAL SMALL BUSINESS CONCERN NONPROFIT ORGANIZATION

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING MIKE DOYLE

TITLE OF PERSON IF OTHER THAN OWNER PRESIDENT
ADDRESS OF PERSON SIGNING 75 SOUTHPARK DRIVE, GLOUCESTER, ONTARIO, CANADA

K1B 3B5

SIGNATURE: 

DATE: 25-JULY-98